1/10

PARTS			SUPPLIERS	
availability	name	supplier id	supplier id	quality
5	PO5	1	1	10
2	PO5	2	2	3
9	PO5	3	3	8

Table and Rank Attributes

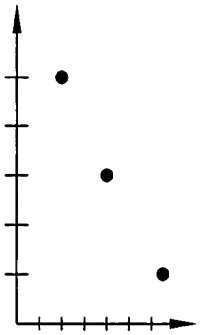
Figure 1

DominatingSet *(C,K)*
Initialize priority queue Q and $D_K = \emptyset$.
Sort the join result in non-increasing order of the s_2 rank value.
For the i th tuple t_i with rank values(s_1^i s_2^i)
    if ($|Q| < K$)
        include t_i in D_K
        include s_1^i in Q
    else
        if $s_1^i \leq \min\{Q\}$ disgard t_i
    else
        include t_i in D_K
        include s_1^i in Q
        if $|Q| > k$ delete the minimum element of Q
Output D_K

The Dominating Set Algorithm

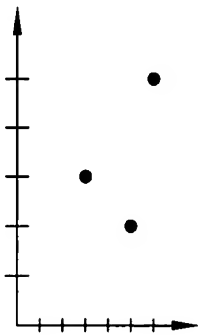
Figure 2

PARTS			SUPPLIER	
availability	name	supplier id	supplier id	quality
5	PO5	1	1	2
3	PO5	2	2	4
1	PO5	3	3	6
			4	1



(a)

PARTS			SUPPLIER	
availability	name	supplier id	supplier id	quality
5	PO5	1	1	5
3	PO5	2	2	3
2	PO5	3	3	4
			4	1



(b)

Examples of set D_K for different join result

Figure 3

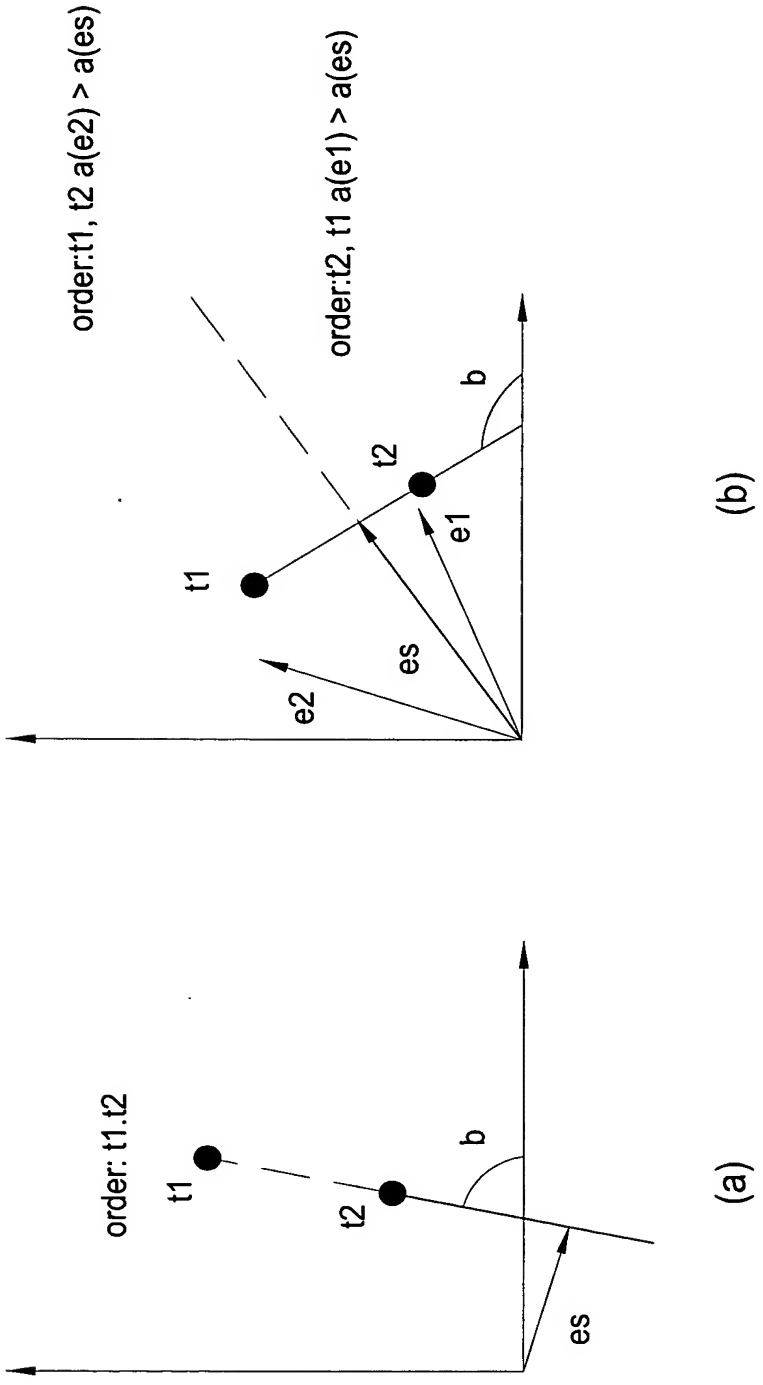
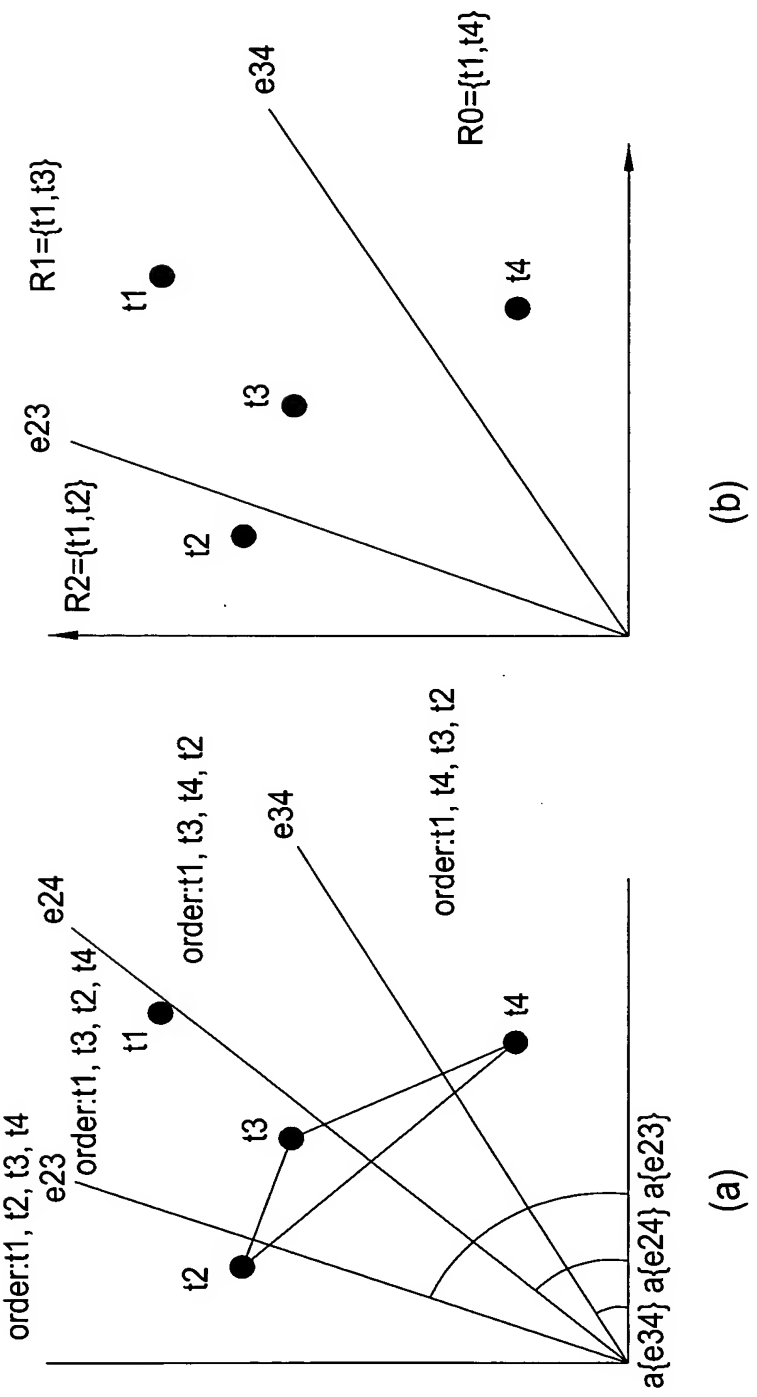

order:t1, t2 a(e2) > a(es)
order:t2, t1 a(e1) > a(es)
b
t2
e1
t1
es
e2
(b)
order: t1.t2
t1
t2
b
es
(a)

Figure 4

ConstructRJI($\mathcal{D}_K$)
For all (t_i, t_j), $t_i, t_j \in \mathcal{D}_K$
  $V \leftarrow$ Compute separating vectors $e_{s_{i,j}}$ and separating points $a(e_{s_{i,j}})$
Sort V in non-decreasing order of $a(e_{s_{i,j}})$
Form R consisting of top-K tuples in $\mathcal{D}_K$ with respect to $f_{(1,0)}$
Set $\ell = 0$; $R_\ell = R$;
For each element (t_i, t_j) of V
    if $t_i, t_j \in R$ or $t_i, t_j \notin R$
        No change in R's composition by $e_{s_{i,j}}$; discard $e_{s_{i,j}}$
    if $t_i \in R$ and $t_j \notin R$
        Materialize $a(e_{s_{i,j}})$, R_ℓ; replace t_i with t_j in R;
        $\ell = \ell + 1$;$R_\ell = R$
    if $t_i \notin R$ and $t_j \in R$
        Materialize $a(e_{s_{i,j}})$,R_ℓ; replace t_j with t_i in R;
        $\ell = \ell + 1$;$R_\ell = R$
When V is exhausted, materialize R

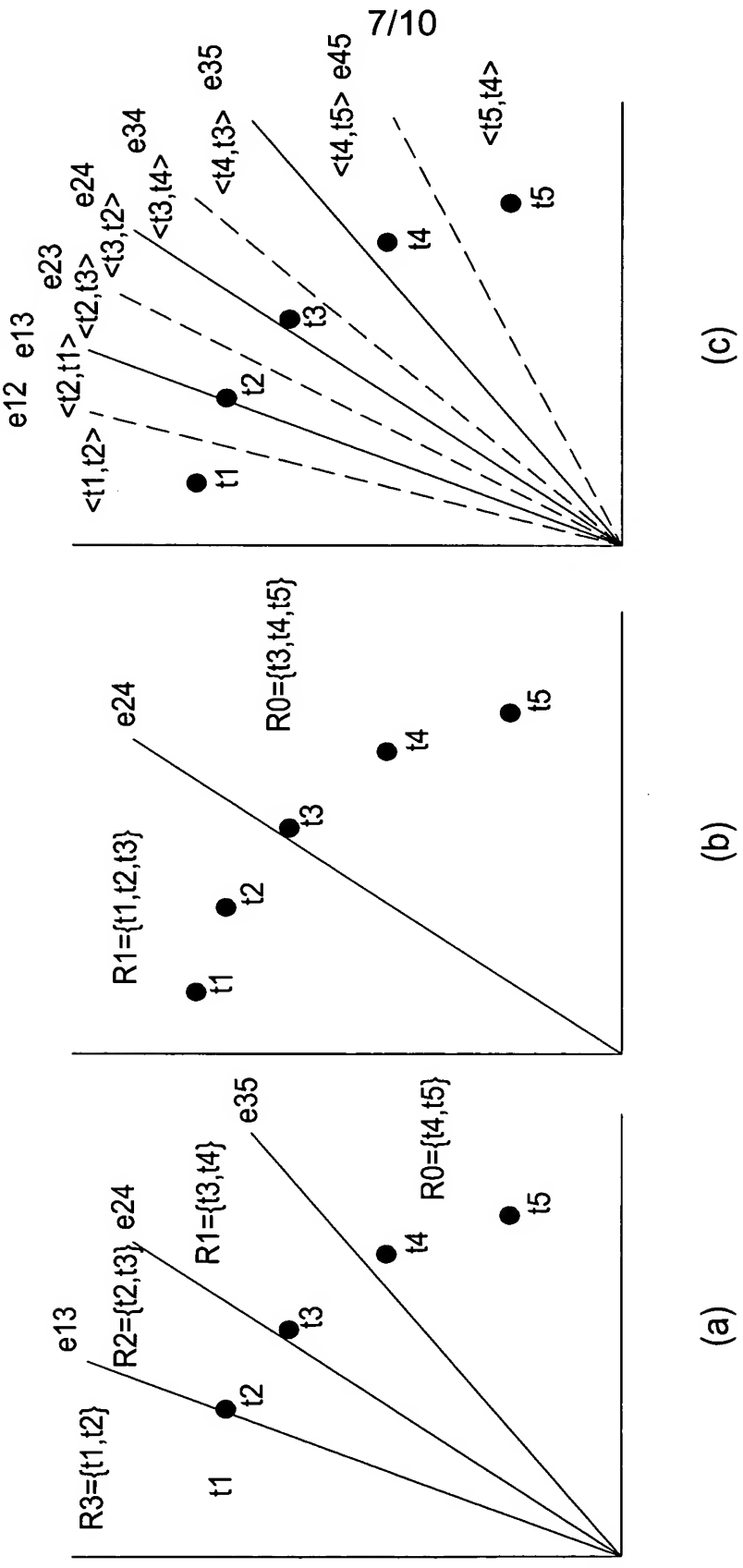
Algorithm ConstructRJI

Figure 5



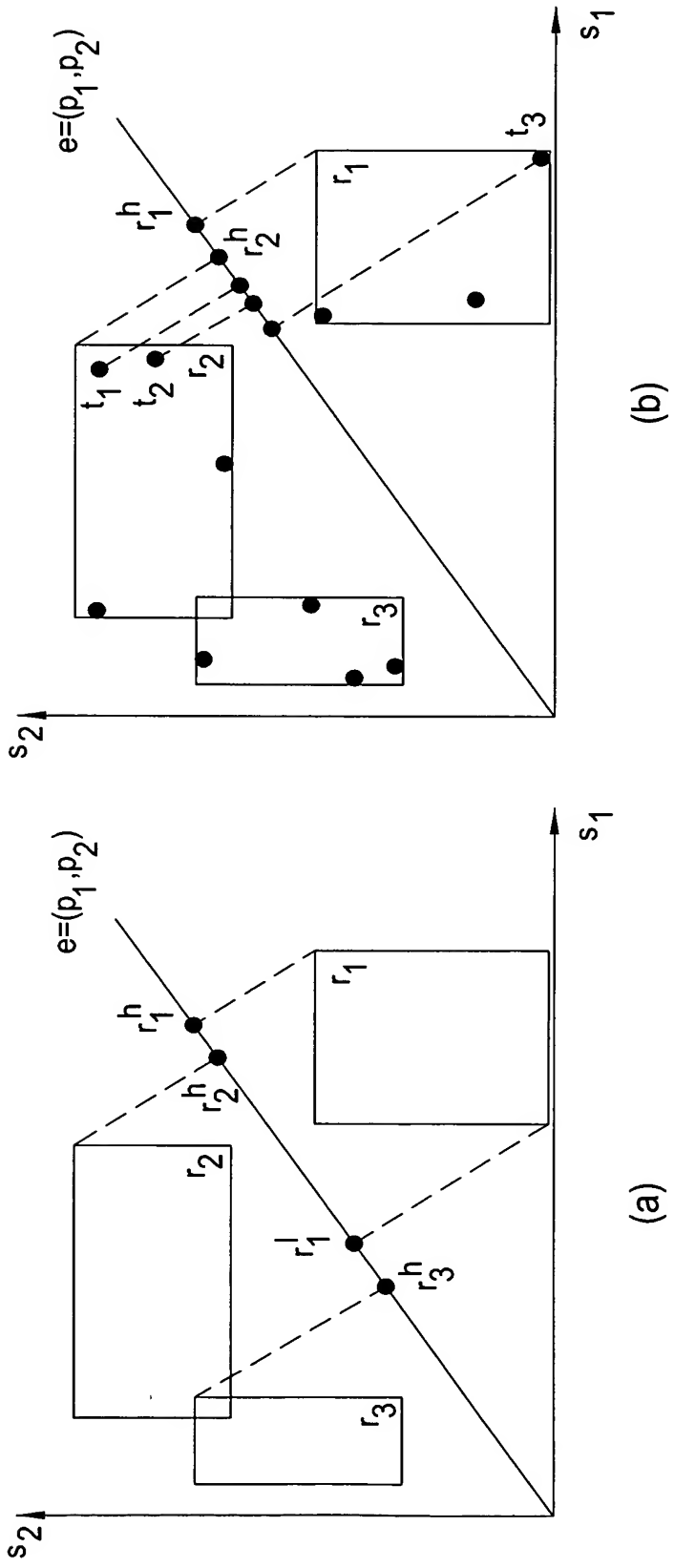
Example operation of algorithm Construct RJI

Figure 6



Space Time Tradeoffs of RJI

Figure 7



A graphical representation for the Top-KrtreeAnswer algorithm

Figure 8

Input: A number k and a preference vector $e = (p_1, p_2)$.
Output: The answer-set S to the *top-k* query.

```
1 procedure TopKrtreeAnswer()
2     let S = ∅ be a priority queue with space for exactly k values;
3     ProcessRtreeNode(root of rtree, S);
4     return(S);

5 procedure ProcessRtreeNode(node N, S)
6     if (N is a leaf)
7         for (all tuples t in this node)
8             insert t in S;
9     else
10        let r range over all the MBRs in N;
11        let r_max = arg max_r {maximum projection of MBR r
                                 on preference vector e};
12        let r_max^low = {minimum projection of MBR r_max
                           on preference vector e};
13        for (each subtree rooted at each MBR c of N)
14            if (maximum projection of MBR c ≥ r_max^low)
15                ProcessRtreeNode(c, S);
16    return(S);
```

The Top KrtreeAnswer algorithm

Figure 9

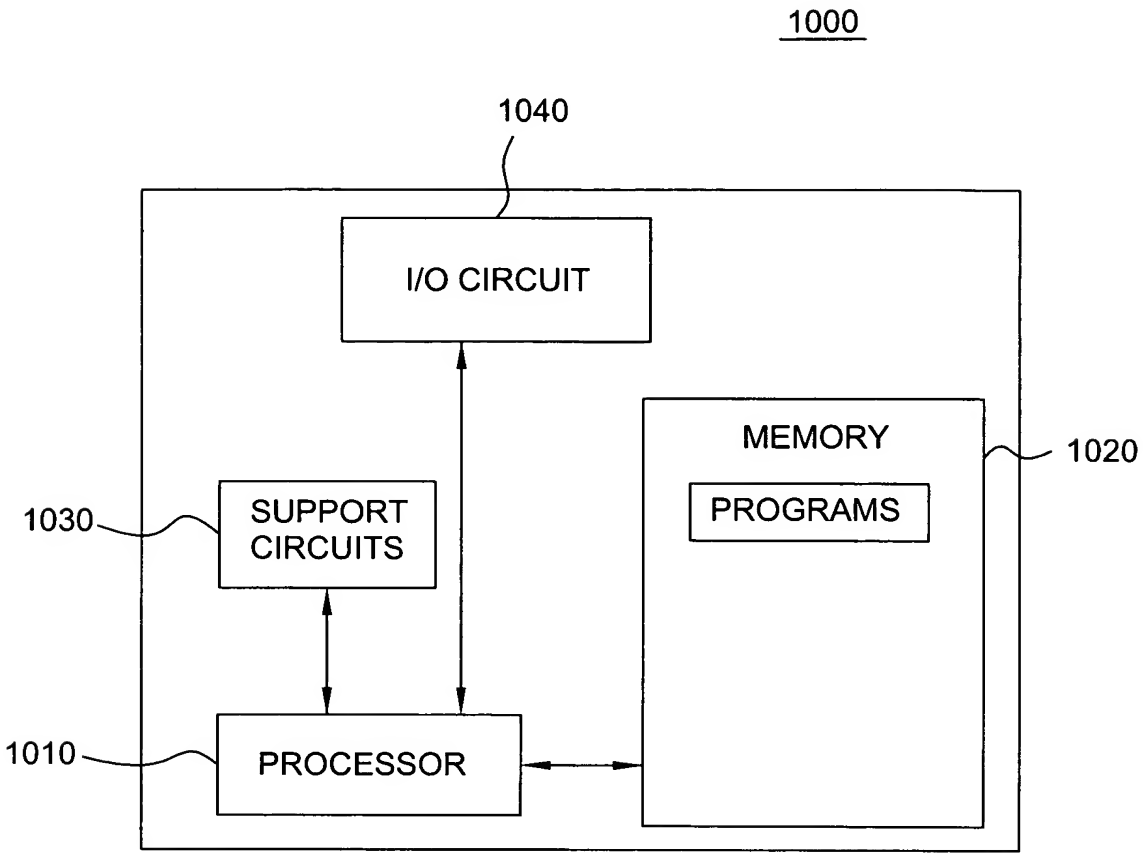

1000
1040
I/O CIRCUIT
MEMORY
PROGRAMS
1020
1030
SUPPORT CIRCUITS
1010
PROCESSOR

FIGURE 10